REMARKS

In the non-final Office Action mailed February 15, 2006, the Examiner noted that claims 52-110 were pending and rejected claims 52-110. Claims 52-110 remain pending for reconsideration which is requested. The Examiner's rejections are traversed below.

On page 2 of the Office Action, the Examiner rejected all claims under 35 U.S.C. § 102 as anticipated by Cozza. On page 2 of the Action the Examiner provided specific comments about the anticipation of claim 52 and on page 3 provided specific comments about the anticipation of claims 53 and 54 over Cozza. However, even though claims 55-110 were mentioned on page 2, no specific comments were provided about claims 55-110. Page 3 of the Office Action rejects all claims under 35 U.S.C. § 103 over Arnold and Cozza. In this rejection, the Examiner also limited the comments that appear on pages 3 and 4 of the Action to claims 52-54 and provided no comments about claims 55-110.

With respect to claims 55-110 in the prior response, the applicant presented specific arguments and traversed the Examiner's rejection particularly noting features of, for example, independent claims 67, 75, 88, 94, 101, 107 and 108 (quarantine), independent claims 72, 79, 84, 85, 92, 97, 98 and 105 (encrypting), independent claim 109 (isolating), and dependent claim 70 (quarantine and encryption). The Examiner has not addressed these features anywhere in this Office Action. Where the applicant traverses a rejection, the Examiner is to take note of the applicant's argument and answer the substance of it (see MPEP 707.07(f)) and any further rejection is to include a rebuttal of any arguments raised in the applicant's reply. (see MPEP 706.07). As a result, the Examiner has not presented a prima facie case of either anticipation or obviousness. Withdrawal of the rejection for this reason is requested.

Further on page 3 of the Action in the anticipation rejection over Cozza, the Examiner appears to include Arnold in the rejection ("As per claims 53 and 54, Arnold discloses ..." see Action page 2, line 6). Clarification of the type (anticipation or obviousness) of rejection of claims 53 and 54 and clarification of which references (Cozza or Cozza and Arnold) are being used in the rejection is requested.

As discussed in the prior response, independent claim 52 emphasizes "a saving unit saving a detected virus-infected file into a specific area within said storage device" (see also claims 57, 62 and 110), independent claims 67, 75, 88, 94, 101, 107 and 108 emphasize that the infected file is quarantined, independent claims 72, 79, 84, 85, 92, 97, 98 and 105 emphasize encrypting an infected file and independent claim 109 emphasizes isolating an infected file from other files.

In contrast, Cozza is directed at:

> The method and apparatus for increasing the speed at which computer viruses are detected stores initial state information concerning the file or volume which is being examined for a virus. This information is stored in a cache in a non-volatile storage medium and when files are subsequently scanned for viruses, the current state information is compared to the initial state information stored in the cache. If the initial state information differs from the current state information then the file or volume is scanned for viruses which change the state information of the file or volume. If the initial state information and current state information is the same then the file or volume is scanned for a subset of viruses which do not change the state information.

(See Cozza Abstract)

> The method and apparatus of the present invention for scanning files for computer viruses relies on the fact that viruses invariably change the file or volume they infect. Consequently, information detailing the initial "state" of an uninfected file or volume can be "cached" or securely saved to disk or other non-volatile storage medium. The cached information is dependent not only on the type of machine the scanning program is running on, but also on viruses' method of infection on that type of machine. The stored information can be tailored to meet the variety of situations found in present and future computing environments.
>
> Once the initial "state" information has been stored to a disk or other non-volatile storage medium, the method and apparatus of the present invention can use this cached information in future virus scans to determine what files and/or volumes have changed in a way indicative of most virus infections. In many applications this information alone is enough to eliminate the need to scan a file/volume for most, if not all, viruses. The result is a substantial improvement in scanning time, in return for a very modest cost in terms of disk or other non-volatile storage medium.

(See Cozza Summary)

> The process for scanning each file in a volume will now be described with reference to FIG. 3. For each file on a volume that is to be scanned, the cache is searched for the presence of the file's cache information in step 40. This is indicated by the presence or absence of the file's file id in the cache (see FIG. 4). Note that if the cache did not exist or if it was invalid, then the file will not be found as the in-memory cache was zeroed. If the file's information is not found (indicating that the file needs to be freshly scanned), then it is scanned for a full complement of viruses, including those that infect the file resource fork in step 42 and those that infect the data fork in step 44.
>
> If the file's scan information is found in the cache then the resource fork length of the file is compared with that stored in the cache in step 46. If the resource fork lengths differ, then the file resource fork has been modified and must be rescanned in step 48 for a full complement of viruses that infect resource forks. If the resource fork size is identical with that stored in the cache, then only a subset of viruses which infect resource forks must be scanned for in step 50. That is, the program must only scan for viruses which infect resource forks but do not change the length of the resource fork, or which have the capability of modifying the scan cache in a attempt to hide themselves. For example, at the present time there

> are no such viruses that affect the resource forks of files on Apple Macintosh computers without changing the resource fork length, so no scanning would be necessary in step 50 if this scanning method is used with an Apple Macintosh computer.
>
> If the file's scan information is found in the cache, then the data fork length of the file is also compared with that stored in the cache. If the data fork length is determined to differ in step 52, then the file data fork has been modified and must be rescanned for a full complement of viruses that infect data forks in step 54. If the data fork size is identical to that stored in the cache, then only a subset of viruses which infect data forks must be scanned for in step 56. Specifically, the program need only scan for viruses which infect data forks but do not change the length of the data fork, or which have the capability of modifying the scan cache in an attempt to hide themselves.
>
> After all virus scanning for a file is completed, the scan cache must be updated. It is preferable to keep a second, new cache in memory separate from the original cache and update that with the new information for each file on the disk (thus eliminating outdated information in the old cache). To update the cache, the scan results are checked to determine whether any virus was found in step 58. If a virus was found, then the scan cache is updated with zeroed information for the file in step 60, which will force the file to be completely scanned again in the future. If no viruses were found in the file, then the file's scan information is added to the new cache in step 62. This information includes the file's ID, resource fork length and data fork length. Steps 38 through 64 are repeated for each scannable file on the disk. When all files have been scanned on the volume, the new, updated cache is written to disk on the volume scanned (34).

(see Cozza, col. 4, line 18-col. 5, line 7)

As can be seen from the above discussion in Cozza, Cozza discusses using a cache information file containing state information about a change in a file to speed up virus scanning. Cozza says nothing about "saving a detected virus-infected file into a specific area within said storage device" as recited in claim 52 much less about the features of independent claims 57, 62 and 110, the features of independent claims 67, 75, 88, 94, 101, 107 and 108, the features of independent claims 72, 79, 84, 85, 92, 97, 98 and 105 and the features of independent claim 109, or about the features of the dependent claims such as claim 70.

As discussed in the prior action, and as acknowledged by the Examiner, Arnold does not teach or suggest the features of the claimed inventions.

It is submitted that the present claims patentably distinguish over Cozza and Cozza with Arnold and withdrawal of the rejection is requested.

It is submitted that the claims are not taught, disclosed or suggested by the prior art. The claims are therefore in a condition suitable for allowance. An early Notice of Allowance is requested.

If any further fees, other than and except for the issue fee, are necessary with respect to this paper, the U.S.P.T.O. is requested to obtain the same from deposit account number 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: August 15, 2006

By: /J. Randall Beckers/
J. Randall Beckers
Registration No. 30,358

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501